REMARKS

Reconsideration and allowance in view of the foregoing amendments and the following remarks are respectfully requested. Specifically, favorable consideration of pending Claims 1-77 is respectfully requested.

As a logistical matter, the Applicant respectfully again repeats its request that the Attorney Docket Number of this application be corrected to be: MS1-282USC6.

Applicant notes that Information Disclosure Statements were filed in this application on July 27, 2000; June 7, 2001; March 13, 2003; and July 19, 2004. Of these, only the initialed and dated PTO-1449s associated with the IDS filed on June 7, 2001 have been received back by Applicant. Applicant respectfully requests that the references listed on the PTO-1449s associated with the IDSs filed on July 27, 2000; March 13, 2003; and July 19, 2004 be considered and that corresponding initialed and dated copies of those PTO-1449s be returned to Applicant.

The amendments to the specification merely update the provenance information relative to the application and/or bring the application into conformance with the presently-employed rules. No new matter is added by the amendments to the specification.

The Rejections Under 35 U.S.C.§103(a)

In the instant Office Action:

I. Claims 1-5, 7, 9-13, 15 and 17-19 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Angelo et al., U.S. Patent No. 5,944,821 (hereinafter "Angelo") in view of Arbaugh (apparently "A Secure and Reliable Bootstrap Architecture, 1996, pp. 1-7; hereinafter "Arbaugh"; the Office Action fails to identify what publication or patent is meant by "Arbaugh", see Office Action, page 3 etc.) and a completely unidentified reference. Clarification of the rejection is respectfully requested.

II. Claims 22, 25, 26, 30-38, 40, 41, 43-54, 56-58, 69, 72, 73, 75 and 76 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Barr et al., U.S. Patent 6,189,100 B1 (hereinafter "Barr") in view of Arbaugh and Angelo.

III. Claims 6, 8, 14, 16 and 21 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Angelo and Arbaugh and further in view of Sadowsky et al., U.S. Patent 6,230,285 (hereinafter "Sadowsky").

IV. Claims 23, 24, 39, 42, 55, 59-62 and 71 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Barr, Arbaugh and Angelo and further in view of Sadowsky.

V. Claim 20 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Arbaugh and Angelo and further in view of LeBourgeois (U.S. Patent 6,026,166 (hereinafter "LeBourgeois").

VI. Claims 63-68, 70, 74 and 77 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Barr, Arbaugh and Angelo and further in view of LeBourgeois.

VII. Claims 27-29 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Barr, Arbaugh, and Angelo, in view of Barlow et al., U.S. Patent 6,038,551 (hereinafter "Barlow").

For at least the reasons that follow, the Applicant respectfully traverses all of the above-listed rejections I through VII, and further requests that these rejections also be reconsidered and withdrawn.

Traverse of Rejections (II), (IV), (VI) and (VII):

Applicant respectfully notes that a portion of the rejections under 35 U.S.C. §103(a), viz., rejections (II), (IV), (VI) and (VII) above, include either Barr and/or Barlow as references. This application was filed on March 10, 1999 and claims priority from Provisional Patent Application Ser. No. 60/105,891, filed on October 26, 1998. This application is assigned to Microsoft Corporation of Redmond WA.

Barr was filed on June 30, 1998 and issued on February 13, 2001. The instant application was filed on March 10, 1999. As such, Barr qualifies as prior art only under the timing provisions of 35 U.S.C. §102(e). Barr is also assigned to Microsoft Corporation of Redmond WA.

Barlow was filed on March 11, 1996 and issued on March 14, 2000. Barlow thus qualifies as prior art only under the timing provisions of 35 U.S.C. §102(e). Barlow is also assigned to Microsoft Corporation of Redmond WA.

Applicant notes the provisions of MPEP §706.02(l)(1), entitled "Rejections Under 35 U.S.C. 102(e)/103; 35 U.S.C. 103(c)". This MPEP section cites 35 U.S.C. §103(c):

> 35 U.S.C. 103. Conditions for patentability; non-obvious subject matter.
> (c) Subject matter developed by another person, which qualifies as prior art only under one or more of subsections (e), (f), and (g) of section 102 of this title, shall not preclude patentability under this section where the subject matter and the claimed invention were, at the time the invention was made, owned by the same person or subject to an obligation of assignment to the same person.

More specifically, this MPEP section states that "Effective November 29, 1999, subject matter which was prior art under former 35 U.S.C. 103 via 35 U.S.C. 102(e) is now disqualified as prior art against the claimed invention if that

subject matter and the claimed invention "were, at the time the invention was made, owned by the same person or subject to an obligation of assignment to the same person." This change to 35 U.S.C. 103(c) applies to all utility, design and plant patent applications filed on or after November 29, 1999, including continuing applications filed under 37 CFR 1.53(b), continued prosecution application filed under 37 CFR 1.53(d), and reissues."

Accordingly, neither Barr nor Barlow is available as prior art under 35 U.S.C. §103(a) with respect to this application, and, as such, rejections (II), (IV), (VI) and (VII) of claims 22-77 are moot. Additionally, no other grounds for rejection have been lodged regarding claims 22-77. Accordingly, in the event that the Examiner still finds such claims to be not allowable, a subsequent non-Final action must be made with different grounds for rejection.

Traverse of Rejection (I):

As noted hereinabove, claims 1-5, 7, 9-13, 15 and 17-19 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Angelo in view of Arbaugh. In traversing the rejections, it is helpful to summarize the teachings of the cited references.

Angelo is directed (Title) to: "Secure software registration and integrity assessment in a computer system". Angelo teaches that: "A method for providing secure registration and integrity assessment of software in a computer system is disclosed. A secure hash table is created containing a list of secure programs that the user wants to validate prior to execution. The table contains a secure hash value (i.e., a value generated by modification detection code) for each of these programs as originally installed on the computer system. This hash table is stored in protected memory that can only be accessed when the computer system is in system management mode. Following an attempt to execute a secured program, a system management interrupt is generated. An SMI handler then generates a current hash value for the program to be executed. In the event that the current hash value matches the stored hash value, the integrity of the program is guaranteed and it is loaded into memory and executed. If the two values do not match, the user is alerted to the discrepancy and may be given the option to update or override the stored hash value by entering an administrative password." (Abstract).

Arbaugh is directed (Title) to: "A Secure and Reliable Bootstrap Architecture". Arbaugh teaches (Abstract) that: "In a computer system, the integrity of lower layers is typically treated as axiomatic by higher layers. Under

the presumption that the hardware comprising the machine (the lowest layer) is valid, integrity of a layer can be guaranteed if and only if: (1) the integrity of the lower layer is checked, and (2) transitions to higher layers occur only after integrity checks on them are complete. The resulting integrity "chain" inductively guarantees system integrity.

When these conditions are met, as they typically are not in the bootstrapping (initialization) of a computer system, no integrity guarantees can be made. Yet, these guarantees are increasingly important to diverse applications such as Internet commerce, security systems, and "active networks." In this paper, we describe the AEGIS architecture for initializing a computer system. It validates integrity at each layer transition in the bootstrap process. AEGIS also includes a recovery process for integrity check failures, and we show how this results in robust systems.

In contrast, claim 1 recites "In a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a software identity register, a method for booting the operating system comprising: computing a cryptographic function of at least a portion of the operating system; and setting the software identity register to a result of the computed cryptographic function", which is not taught, disclosed, suggested or motivated by the cited references, alone or in any proper combination.

Arbaugh does not teach or disclose computing a cryptographic function of at least a portion of the operating system, as recited in claim 1. Arbaugh teaches (p. 68, §3.2.2) verification of Level 4, comprising the operating system, but does not state how this level is verified. Arbaugh teaches (p. 68, §3.2.2, 3RD ¶) that:

"The second section proceeds normally with one change. Prior to executing an expansion ROM, a cryptographic hash is computed and verified against a stored digital signature for the expansion code. If the signature is valid, then control is passed to the expansion ROM. Once the verification of each expansion ROM is complete, (Level 2), the BIOS passes control to the operating system bootstrap code. The bootstrap code was previously verified as part of the BIOS, **and thus no further verification is required**."

Arbaugh then discusses a remainder of the boot and verification processes but is silent with respect to how such boot and verification is carried out. Angelo is directed towards protecting against the execution of unauthorized or modified code, representing application programs, in real time. Angelo describes three embodiments.

The first embodiment relates to a secure hash value being generated for a piece of software when it is installed on a computer system (Angelo, col. 4, lines 45-48), and the second embodiment relates to a secured hash value for a table being maintain in a system management mode memory. Notably, Angelo acknowledges that, "Both of the aforementioned embodiments of the invention have the additional advantage of being operating system independent," (Angelo, col. 5, lines 17-19).

The third embodiment, which is therefore the only embodiment that is contextually relevant to the rejected claims, "builds on a trusted boot facility," (Angelo, col. 5, lines 25-29), meaning that such embodiment activates after the operating system is booted.

The Office Action states (p. 4) that Angelo does not disclose computing a cryptographic function of at least a portion of the operating system and setting the software identity register to a result of the computed cryptographic function. The Office Action nowhere identifies any portion of Arbaugh that might include setting any software identity register to any result, or to any result of a computed cryptographic function. As a result, the rejection of claim 1 based on Angelo in view of Arbaugh fails to provide the elements recited in claim 1.

Furthermore, Arbaugh describes a boot procedure for ensuring a valid and secure starting point for operation of computer systems. Arbaugh thus is concerned with boot operation only.

In contrast, Angelo describes validation of application programs during system operation. Angelo states (Summary, col. 4, line 26 et seq.) that: "A computer system according to the present invention incorporates the capability to protect against the execution of unauthorized or modified code in real time, **as opposed to relying solely on power-up routines to maintain a secure and trusted path**."

Angelo thus teaches away from the disclosure of Arbaugh. It is improper to combine references that teach away from one another. This is explained below in more detail with reference to MPEP §2145, entitled "Consideration of Applicant's Rebuttal Arguments". In a subsection (X)(D)(2), entitled "References Cannot Be Combined Where Reference Teaches Away from Their Combination", this MPEP section states that: "It is improper to combine references where the references teach away from their combination. *In re Grasselli*, 713 F.2d 731, 743, 218 USPQ 769, 779 (Fed. Cir. 1983)".

Accordingly, the combination of references proposed in the Office Action is improper.

One reason that Angelo is concerned with being able to verify the integrity of **application programs** prior to execution thereof but after booting is because of ongoing risk of file corruption during system operation via viruses (see, e.g., col. 2, line 4 through line 45; col. 5, line 20 et seq.; col. 9, line 26 et seq.; col. 11, line 51 et seq.). Modification of the teachings of Angelo to perform the bootstrap operations of Arbaugh renders the teachings of Angelo unsuitable for their intended purpose, viz., to detect viral infection of application programs as they are invoked an dafter system booting.

Modification of the teachings of a reference in such a way as to render those teachings unsuitable for their intended purpose renders motivation to make such modification nugatory **as a matter of law**. Such is explained below in more detail with reference to MPEP §2143.01, entitled "Suggestion or Motivation to Modify the References". In a subsection entitled "THE PROPOSED MODIFICATION CANNOT RENDER THE PRIOR ART UNSATISFACTORY FOR ITS INTENDED PURPOSE", this MPEP section states that: "If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984)".

Accordingly, there is no motivation to modify the teachings of Angelo as suggested in the Office Action.

Claim 3 recites "In a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a software identity register, a method for booting the operating system comprising: executing an atomic operation to set an identity of the operating system into the software identity register of the CPU, wherein in an event that the atomic operation completes correctly, the software identity register contains the identity of the operating system and in an event that the atomic operation fails to complete correctly, the software identity register contains a value other than the identity of the operating system; and examining a content of the software identity register to verify the identity of the operating system", while claim 11 recites "In a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a software identity register, a method comprising: identifying a boot block of code in the OS that uniquely describes the OS; creating an identity of the OS from the boot block; and executing an atomic operation to set the identity of the operating system into the software identity register of the CPU, wherein in an event that the atomic operation completes correctly, the software identity register contains the identity of the operating system", which recitations are not taught, disclosed, suggested or motivated by the cited references.

Angelo is silent regarding any atomic operation. In fact, Angelo is void of the word "atomic". Notably, the Office Action similarly is silent with respect to this affirmatively-recited aspect of the subject matter recited in claims 3 and 11.

The term "atomic" as applied to operation of a computer is a term of art. This term of art is defined, inter alia, in the Microsoft Press Computer

Dictionary, Third Ed., published by Microsoft Press, a division of Microsoft Corporation, One Redmond Way, Redmond WA (copyright 1997) at page 34. That definition appears hereafter: an atomic Operation is "An operation considered or guaranteed to be indivisible (by analogy with an atom of matter, once thought to be indivisible). Either the operation is uninterruptable or, if it is aborted, a mechanism is provided that ensures the return of the system to its state prior to initiation of the operation."

Angelo, as noted above, is void of the term "atomic" and also is void of any mention of any uninterruptable operation. As such, it is inconceivable that Angelo could teach, disclose, suggest or motivate the subject matter recited in these claims.

Claim 19 recites "In a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a pair of private and public keys and a software identity register that holds an identity of the operating system, a method comprising: creating an identity of the OS containing the identity from the software identity register, information describing the operating system, and the CPU public key; and signing the OS certificate using the CPU private key", which is not taught, disclosed, suggested or motivated by the cited references.

The Office Action states (p. 7) that Angelo teaches "the CPU having a software identity register (Fig. 2 in combination with column 9 lines 35-38)." Applicant disagrees. The cited passage states "As used in this disclosure, the term "secure hash value" or "hash value" refers generally to a value--generated

by an integrity assessment code--that is specific to a given software **application**".

The Office Action then states (p. 7) that "In addition, Angelo discloses a system wherein in an event that the atomic operation completes correctly, the software identity register contains the identity of the operating system (column 10 lines 16-28) and in an event that the atomic operation fails to complete correctly, the software identity register contains a value other than the identity of the operating system; and examining a content of the software identity register to verify the identity of the operating system (column 10, lines 39-65). The hash value can be deleted; this would be setting the value to something other than the correct hash value. The user is also given a choice to update the value and put in a value that is different from the correct hash value." Applicant disagrees for at least the following three reasons.

First, the Office Action fails to accurately report what the reference states. Angelo states (col. 10, lines 13-65) that:

> As part of step 308, the SMI handler 202 next determines if the hash table 206 contains a hash value corresponding to the [application] program to be executed. Typically, a secure hash value is created for each program to be tracked as part of the program's installation into the computer system S. If a hash value for the program is found, control proceeds to step 310 where the stored hash value is retrieved. Control then proceeds to step 312 for a comparison of the newly generated hash value with the stored hash value. If the two values are the same, control passes to step 318 and the program is loaded into memory and executed. As mentioned, the program or portions of it can be loaded into SMM memory 200 for execution. For example, if the user is performing encryption, it would not be desirable to have the encryption algorithm or password exposed in normal memory. Alternatively, the program can be executed from normal memory if secure execution is not needed. In the later case, system management mode can be exited prior to execution of the program. In either case, control then proceeds to step 320 where the relevant memory is cleaned up. Control next

passes to step 324 and system management mode is exited (assuming system management mode was not exited at an earlier point).

If no hash value corresponding to the program to be executed is found as a result of step 308, control proceeds to step 314 and the user is informed that the program is not properly registered to be executed. Control then proceeds to step 316, which is also where control proceeds if the stored hash value does not equal the newly calculated hash value as determined in step 312 (for example, the program has been modified by a virus or a new version of the program has been installed). In step 316, which is an optional enhancement to the invention, the system is configured to query the user to update the hash table 206 and/or stored hash value to incorporate the program as it currently exists. Alternatively, the user could simply be, asked for permission to run the program in its altered state.

If the user desires to update the hash table 206, control passes to step 322 and the subroutine UPDATE 400 (FIG. 4) is called. Following a return from UPDATE 400, or if UPDATE 400 is not called following step 316, control passes to step 324 and the processor 102 exits system management mode.

Referring now to FIG. 4, a flowchart illustration of a secure method UPDATE 400 for updating a stored hash table or stored hash value is shown. In addition to adding or updating entries for programs that the user wants to verify prior to execution, entries can be deleted for programs that are no longer utilized. The subroutine UPDATE 400 is called in step 322 of FIG. 3.

This passage is void of any mention of atomic execution of anything. This passage is also void of any mention whatsoever of any operating system. This passage also does not provide any teaching or disclosure of "creating an identity of the OS containing the identity from the software identity register, information describing the operating system, and the CPU public key; and signing the OS certificate using the CPU private key", as recited in claim 19. This passage discusses storage of hash values for **application programs**.

As stated beginning at col. 7, line 43, the system management interrupt function was developed responsive to power conservation needs in portable

computers. A SMI call results in the present state of the processor being stored in the system management memory of Fig. 2. A power management function, like an application program, is, by definition, invoked only **after** successfully having booted the computer, that is, after the operating system had been successfully activated. There is thus no need whatsoever to employ the SMM as suggested in the Office Action in the context of the teachings of Angelo.

Second, the claim recites "creating an identity of the OS containing the identity from the software identity register, information describing the operating system, and the CPU public key; and signing the OS certificate using the CPU private key". There is no discussion whatsoever in Angelo of operating system identity. There is no discussion whatsoever in Angelo of creation of OS identity, or of doing so using information describing the OS or any public key or of signing any certificate using a private key.

In fact, Angelo is void of the term "public key" and uses the Hollerith string "private key" only in the context of reference (col. 11, lines 23-30) to "An improved method for establishing a secure keyboard link to enter password and other information is disclosed in U.S. Pat. No. 5,748,888 entitled "METHOD AND APPARATUS FOR PROVIDING SECURE AND PRIVATE KEYBOARD COMMUNICATIONS IN COMPUTER SYSTEMS", filed May 29, 1996 and hereby incorporated by reference."

Third, an application program is not an operating system, and an operating system is not an application program. Conflating the two terms gives each a meaning repugnant to the ordinary meaning of the term. It is improper to employ terminology in a manner that is repugnant to the ordinary meaning of the

terminology, as is explained below in more detail with reference to MPEP §2111.01, entitled "Plain Meaning". This MPEP section states that "THE WORDS OF A CLAIM MUST BE GIVEN THEIR "PLAIN MEANING" UNLESS THEY ARE DEFINED IN THE SPECIFICATION", and that "While the meaning of claims of issued patents are interpreted in light of the specification, prosecution history, prior art and other claims, this is not the mode of claim interpretation to be applied during examination. During examination, the claims must be interpreted as broadly as their terms reasonably allow. This means that the words of the claim must be given their plain meaning unless applicant has provided a clear definition in the specification. *In re Zletz*, 893 F.2d 319, 321, 13 USPQ2d 1320, 1322 (Fed. Cir. 1989) ...."

This principle of interpretation is explained more fully in MPEP §608.01(o), entitled "Basis for Claim Terminology in Description". This MPEP passage states that:

> The meaning of every term used in any of the claims should be apparent from the descriptive portion of the specification with clear disclosure as to its import; and in mechanical cases, it should be identified in the descriptive portion of the specification by reference to the drawing, designating the part or parts therein to which the term applies. A term used in the claims may be given a special meaning in the description. No term may be given a meaning repugnant to the usual meaning of the term.

Dictionary definitions of these terms are given below, taken from the Microsoft Press Computer Dictionary, Third Ed., published by Microsoft Press, a division of Microsoft Corporation, One Redmond Way, Redmond WA (copyright 1997).

An **application** program is (p. 27) "A program designed to assist in the performance of a specific task, such as word processing, accounting, or inventory

management." In contrast, an **operating system** is (p. 341) "The software that controls the allocation and usage of hardware resources such as memory, central processing unit (CPU) time, disk space, and peripheral devices. The operating system is the foundation on which **applications** are built. Popular operating systems include Windows 95, Windows NT, Mac OS, and UNIX."

As such, operating systems and application programs are different, are not arbitrarily interchangeable and improper and inappropriate usage of one term for the other or vice versa gives both terms meanings repugnant to the ordinary meanings of these terms.

The rejection fails to meet appropriate standards for a finding of unpatentability, as noted above and also below. For at least these reasons, the rejection of claims 1, 3, 11 and 19, and claims dependent therefrom, is improper and should be withdrawn, and claims 1, 3, 11 and 19 and claims dependent therefrom should be allowed.

Traverse of Rejection (III):

As noted hereinabove, claims 6, 8, 14, 16 and 21 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Angelo and Arbaugh and further in view of Sadowsky.

Sadowsky is directed (Title) to: "Boot failure recovery". Sadowsky teaches: "A boot failure recovery system operates to diagnose a failed system boot in a computer operating system which boots by bootstrapping from a boot sector (12) of a storage medium (10) using configuration information (82). The boot failure recovery system includes an agent (24) which monitors operating system files used during system boot and which stores information regarding changes to the system files to a change file. A repair module (22) analyzes the change file to determine the cause of the failed system boot. A boot check module (16) responds to initiation of a system boot by determining if a prior system boot was successful. Boot check module (16) causes execution of a first boot sector code module (16) upon occurrence of a successful prior system boot and causes execution of the repair module (22) upon occurrence of a failed prior system boot." (Abstract).

In contrast, claims 6, 8, 14 and 16 each recite "appending at least a portion of the identity to a boot log", claims 8 and 16 further recite "authenticating additional blocks of code; and appending identities of the additional blocks of code to the boot log" and claim 21 recites "wherein creating an identity of the OS comprises forming the OS certificate with one or more items from a boot log containing identities of software components that are executing on the CPU", which recitations are not taught, disclosed, suggested or motivated by the cited

references. As noted above, Arbaugh and Angelo do not provide the elements recited in the base claims. Sadowsky fails to cure these deficiencies.

The Office Action states (p. 14) that "Sadowsky discloses maintaining a boot log (Fig. 4). Further Sadowsky suggest [sic] the boot file comprising appending at least a portion of the identity to a boot log (column 4 lines 65 and 66)." The Office Action is indefinite as to the identity allegedly referenced by Sadowsky. Applicant finds no mention of identity of any operating system in the context of a boot log in Sadowsky. Clarification of the rejection is respectfully requested.

Sadowsky describes the bootlog.txt file as containing information relevant to events occurring during boot (col. 4, lines 61-64), with no suggestion that the identity of the operating system constitutes an "event" occurring during boot. Further, Sadowsky determines the cause of boot failure taking the chronology of boots into consideration, and there is not even an implied suggestion that the operating system identity is desirable for such determination.

The rejection fails to meet appropriate standards for a finding of unpatentability, as noted above and also below. Accordingly, the rejection of claims 6, 8, 14, 16 and 21 is prima facie defective and should be withdrawn, and claims 6, 8, 14, 16 and 21 should be allowed.

Traverse of Rejection (V):

As noted previously, claim 20 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Arbaugh and Angelo and further in view of LeBourgeois.

LeBourgeois is directed to (Title): "Digitally certifying a user identity and a computer system in combination". LeBourgeois teaches a: "Digital certification method in which a first digital signature dependent upon a first user identity and a first user system in combination, is stored accessibly to a certification server. The first user identity can be distinguished by, for example, a PIN provided by the user. Subsequently, the user system generates a second signature dependent upon both the current user identity and the current user system in combination. The certifying system then compares the second signature with the first, as stored, to certify the transaction. The certification can accommodate normal computer system component drift. An inquiring system, desiring to confirm the identity of a user, issues a challenge code to the user system. The user system then digests the user's PIN, individual component signatures as they currently exist on the user's system, together with the challenge code to generate the new signature. The new signature is transmitted back to the inquiring system, which transmits it on to the certification server together with the challenge code. The certification server then digests the challenge code with the original signature as previously stored, and compares the result to the newly provided signature to confirm the users identity, else drift criteria can be applied if desired." (Abstract).

In contrast, claim 20 recites "submitting the signed OS certificate over a network to a third party to prove an identity of the operating system to the third party", which is not taught, disclosed, suggested or motivated by the cited references. As noted above, Angelo and Arbaugh fail to provide the elements of claim 19, from which claim 20 depends. LeBourgeois fails to cure the deficiencies of Angelo and Arbaugh.

Angelo and Arbaugh fail to provide any signed OS certificate. The passing mention of the digital product right management technique in LeBourgeois has no relevance to proving operating system identity to a third party; instead it (col. 3, lines 21-23) ensures that "digital products, such as software, music, images and so on, be authorized for use only on a single computer."

The rejection fails to meet appropriate standards for a finding of unpatentability, as noted above and also below. Accordingly, the rejection of claim 20 is in error and should be withdrawn, and claim 20 should be allowed.

All of the rejections based on combinations of elements taken from Angelo and Arbaugh fail to meet the standards for a finding of unpatentability set forth in MPEP §2143, entitled "Basic Requirements of a Prima Facie Case of Obviousness" (see also MPEP §706.02(j), §2141 et seq. and §2142). Further, simply providing a conclusory statement that "It would have been obvious ...." fails to meet the standards set forth in the MPEP for establishing a prima facie case of unpatentability. These are set forth in MPEP §2143, entitled "Basic Requirements of a Prima Facie Case of Obviousness" (see also MPEP §706.02(j), §2141 et seq. and §2142).

This MPEP section states that "To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings." The references fail to teach or disclose the elements recited in the claims, as noted with specificity hereinabove. Accordingly, the references cannot possibly provide motivation to modify their teachings to arrive at the invention as claimed, and the Examiner has identified no such teaching or disclosure in the references. As a result, the first prong of the test cannot be met.

MPEP §2143 further states that "Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations."

Inasmuch as the references fail to provide all of the features recited in Applicant's claims, as noted with specificity hereinabove, the third prong of the test is not met. As a result, there cannot be a reasonable expectation of success. As such, the second prong of the test cannot be met.

MPEP §2143 additionally states that "The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)." This fourth criterion cannot be met because the references fail to teach or disclose the elements recited in the claim. As such, the unpatentability rejections fail all of the criteria for establishing a prima facie case of obviousness as set forth in the MPEP.

Arguendo, even if Arbaugh, and Angelo could be combined, the resultant combination would not be obvious because the prior art does not suggest the desirability of the combination due to their contextual disparity, *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990), as discussed in MPEP §2143.01.

Further, the Office Action identifies no teaching whatsoever in Angelo or Arbaugh of the subject matter recited in these claims. Additionally, there is no teaching or disclosure, or guidance, suggestion or motivation identified in the references or by the Office Action to attempt to combine or modify, or to aid one of ordinary skill in picking and choosing elements from the diverse embodiments of the references or in assembling those elements to attempt to arrive at the subject matter of any of Applicant's claims. As such, the rejection appears to employ an inappropriate 'obvious to try' standard of unpatentability.

Such is improper, as is discussed below in more detail with reference to MPEP §2145(X)(B), entitled "Obvious To Try Rationale". This MPEP section states that "The admonition that 'obvious to try' is not the standard under §103 has been directed mainly at two kinds of error. In some cases, what would have been 'obvious to try' would have been to vary all parameters or try each of numerous possible choices until one possibly arrived at a successful result, where the prior art gave either no indication of which parameters were critical or no direction as to which of many possible choices is likely to be successful. In others, what was 'obvious to try' was to explore a new technology or general approach that seemed to be a promising field of experimentation, where the prior art gave only general guidance as to the particular form of the claimed invention

or how to achieve it." *In re O'Farrell*, 853 F.2d 894, 903, 7 USPQ2d 1673, 1681 (Fed. Cir. 1988) (citations omitted)".

In this instance, no guidance in selecting some but not others of the many elements from the many embodiments of the references is identified. Similarly, no direction as to which of many possible choices is likely to be successful has been identified.

As there is no basis for the Examiner's contentions within the cited references, the only possible motivation for these contentions is hindsight reconstruction wherein the Examiner is utilizing Applicant's own disclosure to construct a reason for combining and/or modifying the teachings of the cited references. The Examiner is reminded that hindsight reconstruction is not an appropriate basis for a §103 rejection. (See, e.g., *Interconnect Planning Corp. v. Feil*, 227 USPQ 543, 551 (Fed. Cir. 1985); *In re Mills*, 16 USPQ2d 1430 (Fed. Cir. 1990) (explaining that hindsight reconstruction is an improper basis for rejection of a claim).)

Additionally, no **evidence** has been provided as to why it would be obvious to combine or modify the teachings of these references. Evidence of a suggestion to combine or modify may flow from the prior art references themselves, from the knowledge of one skilled in the art, or from the nature of the problem to be solved. However, this range of sources does not diminish the requirement for actual evidence. Further, the showing must be clear and particular. See *In re Dembiczak*, 175 F.3d 994, 998 (Fed. Cir. 1999).

Conclusion

To recapitulate, rejections (II), (IV), (VI) and (VII) are moot. The remaining rejections (i) fail to comport with the requirements for an unpatentability rejection, (ii) give terms used in the references meanings repugnant to the ordinary meanings of the terms, (iii) employ an improper "obvious to try" standard for a finding of unpatentability (iv) using hindsight reconstruction and (v) lack any showing of proper evidence of motivation to combine. Further, (vi) the references teach away from each other and from the claimed subject matter and (vii) essential purposes of the references are destroyed in modifying the references as suggested in the Office Action.

All objections and rejections having been addressed, it is respectfully submitted that the present application is now in condition for allowance. Early and forthright issuance of a Notice of Allowability is respectfully requested.

Respectfully Submitted,

Lee & Hayes, PLLC

Dated: Oct. 19, 2004

Frederick M. Fliegel
Reg. No. 36,138
(509) 324-9256 x239

Lee & Hayes, PLLC
421 W. Riverside
Suite 500
Spokane, WA 99201